

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A method of software loading and initialization in a distributed network of nodes, the method comprising ~~the computer-implemented steps of:~~
~~providing a master node;~~
~~providing software package storage means on said master node for storing, in a first~~
~~storage of a master node, software packages and boot images, which software~~
~~packages and boot images will be used by [[that]] the nodes in the distributed~~
~~network will be using as well as older versions that are kept for regressing a~~
~~node back to a previous boot image or software package version;~~
~~providing node information storage means on said master node for storing, in a~~
~~second storage of the master node, preferred software version information[[,]]~~
~~and node type, and other pertinent information for each node in the distributed~~
~~network;~~
~~receiving, at the master node, a request for a boot image and software packages from~~
~~a node, in the distributed network, that is wherein a node performing an initial~~
~~boot requests a boot image and software packages from said master node;~~
~~based on the request, retrieving said node's preferred software version information of~~
~~the node from said node information storage means the second storage;~~
~~using the preferred software version information of the node, extracting a boot image~~
~~and one or more software packages from the first storage said software~~
~~package storage means using said node's preferred software version~~
~~information;~~
~~delivering, to the node, the extracted boot image and one or more software packages~~
~~to said node;~~
wherein ~~[[said]]~~ the node stores the extracted boot image and one or more software
packages in its local persistent storage; ~~[[and]]~~
wherein software version information is extracted from the one or more software
packages and stored in the local persistent storage; and

wherein ~~[[said]]~~ the node reboots and executes the boot image stored in the local persistent storage.

2. (currently amended) A method as recited in Claim 1, wherein said node, based on a command from said master node, does not store the one or more software packages in the local persistent storage device, allowing said master node to download test software packages to said node and temporarily run said node using the test software packages, and wherein when said node reboots, the test software packages will no longer exist on said node.
3. (currently amended) A method as recited in Claim 1, wherein ~~[[said]]~~ retrieving ~~[[step]]~~ preferred software version information creates ~~said node's~~ the preferred software version information from ~~said node information storage means~~ the second storage based on functional features requested by said node.
4. (original) A method as recited in Claim 1, wherein said node verifies the software version information with said master node.
5. (original) A method as recited in Claim 4, wherein if said node has the correct software versions, then said node completes booting by executing the software packages stored in the local persistent storage.
6. (currently amended) A method as recited in Claim 4, ~~wherein~~ further comprising, if said node does not have the correct software versions, ~~said master node retrieves~~ retrieving correct software packages from ~~said software package storage means~~ the first storage and ~~[[sends]]~~ sending the correct software packages to said node, ~~[[and]]~~ wherein said node stores the correct software packages in the local persistent storage and completes booting by executing the correct software packages stored in the local persistent storage.

7. (original) A method as recited in Claim 1, wherein the master node has the ability to categorize nodes into classes where all of the nodes in a particular class of nodes have the same software configuration and may have differing processor types.
8. (original) A method as recited in Claim 1, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.
9. (original) A method as recited in Claim 1, wherein a boot image is customized for a particular type of node and provides basic low-level communications.
10. (currently amended) A method of software loading and initialization in a distributed network of nodes, the method comprising ~~the computer implemented steps of:~~
providing a master node;
providing software package storage means on said master node for storing, in a first storage of a master node, software packages and boot images, which software packages and boot images will be used by [[that]] the nodes in the distributed network will be using as well as older versions that are kept for regressing a node back to a previous boot image or software package version;
providing node information storage means on said master node for storing, in a second storage of the master node, preferred software version information[[,]] and node type, and other pertinent information for each node in the distributed network;
receiving, at the master node, a request for a boot image and software packages from a node, in the distributed network, that is wherein a node performing an initial boot requests a boot image and software packages from said master node;
based on the request, retrieving said node's preferred software version information of the node from said node information storage means the second storage;
using the preferred software version information of the node, extracting a boot image and one or more software packages from the first storage said software

~~package storage means using said node's preferred software version~~
~~information;~~

~~delivering, to the node,~~ the extracted boot image and one or more software packages
~~to said node.~~

11. (currently amended) A method as recited in Claim 10, wherein said node stores the extracted boot image and one or more software packages in its local persistent storage and wherein software version information is extracted from the one or more software packages and stored in the local persistent storage.
12. (currently amended) A method as recited in Claim ~~[[11]]~~ 10, wherein said node, based on a command from said master node, does not store the one or more software packages in ~~[[the]]~~ its local persistent storage device, allowing said master node to download test software packages to said node and temporarily run said node using the test software packages, and wherein when said node reboots, the test software packages will no longer exist on said node.
13. (original) A method as recited in Claim 11, wherein said node reboots and executes the boot image stored in the local persistent storage, and wherein said node verifies the software version information with said master node.
14. (original) A method as recited in Claim 13, wherein if said node has the correct software versions, then said node completes booting by executing the software packages stored in the local persistent storage.
15. (currently amended) A method as recited in Claim 13, further comprising, ~~wherein~~ if said node does not have the correct software versions, ~~said master node retrieves~~ retrieving correct software packages from ~~said software package storage means~~ the first storage and ~~[[sends]]~~ sending the correct software packages to said node, ~~[[and]]~~ wherein said node stores the correct software packages in the local persistent storage

and completes booting by executing the correct software packages stored in the local persistent storage.

16. (original) A method as recited in Claim 10, wherein the master node has the ability to categorize nodes into classes where all of the nodes in a particular class of nodes have the same software configuration and may have differing processor types.
17. (original) A method as recited in Claim 10, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.
18. (original) A method as recited in Claim 10, wherein a boot image is customized for a particular type of node and provides basic low-level communications.
19. (currently amended) A method as recited in Claim 10, further comprising:
executing wherein a user installs a composite image, that is installed by a user onto
said master node, ~~which is executed and creates~~ to create boot images,
software packages, and node information[[,]]; and
~~wherein said master node places~~ placing the boot images and software packages in
~~said software package storage means~~ the first storage and the node
~~information in said node information storage means~~ the second storage.
20. (currently amended) A method as recited in Claim 10, wherein [[said]] retrieving step
preferred software version information creates ~~said node's~~ the preferred software
version information from ~~said node information storage means~~ the second storage
based on functional features requested by said node.
21. (currently amended) A computer-readable medium carrying one or more sequences of
instructions for software loading and initialization in a distributed network of nodes,
which instructions, when executed by one or more processors, cause the one or more
processors to ~~carry out the steps of~~ perform:

~~providing a master node;~~

~~providing software package storage means on said master node for storing, in a first storage of a master node, software packages and boot images, which software packages and boot images will be used by [[that]] the nodes in the distributed network will be using as well as older versions that are kept for regressing a node back to a previous boot image or software package version;~~

~~providing node information storage means on said master node for storing, in a second storage of the master node, preferred software version information[[,]] and node type, and other pertinent information for each node in the distributed network;~~

~~receiving, at the master node, a request for a boot image and software packages from a node, in the distributed network, that is wherein a node performing an initial boot requests a boot image and software packages from said master node;~~

~~based on the request, retrieving said node's preferred software version information of the node from said node information storage means the second storage;~~

~~using the preferred software version information of the node, extracting a boot image and one or more software packages from the first storage said software package storage means using said node's preferred software version information;~~

~~delivering, to the node, the extracted boot image and one or more software packages to said node.~~

22. (currently amended) A computer-readable medium as recited in Claim 21, wherein said node stores the extracted boot image and one or more software packages in its local persistent storage and wherein software version information is extracted from the one or more software packages and stored in the local persistent storage.
23. (currently amended) A computer-readable medium as recited in Claim ~~[[22]]~~ 21, wherein said node, based on a command from said master node, does not store the one or more software packages in ~~[[the]]~~ its local persistent storage device, allowing said master node to download test software packages to said node and temporarily run

said node using the test software packages, and wherein when said node reboots, the test software packages will no longer exist on said node.

24. (original) A computer-readable medium as recited in Claim 22, wherein said node reboots and executes the boot image stored in the local persistent storage, and wherein said node verifies the software version information with said master node.
25. (original) A computer-readable medium as recited in Claim 24, wherein if said node has the correct software versions, then said node completes booting by executing the software packages stored in the local persistent storage.
26. (currently amended) A computer-readable medium as recited in Claim 24, ~~wherein~~ further comprising, if said node does not have the correct software versions, ~~said master node retrieves~~ retrieving correct software packages from ~~said software package storage means~~ the first storage and ~~[[sends]]~~ sending the correct software packages to said node, ~~[[and]]~~ wherein said node stores the correct software packages in the local persistent storage and completes booting by executing the correct software packages stored in the local persistent storage.
27. (original) A computer-readable medium as recited in Claim 21, wherein the master node has the ability to categorize nodes into classes where all of the nodes in a particular class of nodes have the same software configuration and may have differing processor types.
28. (original) A computer-readable medium as recited in Claim 21, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.
29. (original) A computer-readable medium as recited in Claim 21, wherein a boot image is customized for a particular type of node and provides basic low-level communications.

30. (currently amended) A computer-readable medium as recited in Claim 21, ~~wherein a user installs~~ further comprising:
executing a composite image onto said master node which is executed and creates ,
that was installed by a user, to create boot images, software packages, and
node information[[,]]; and
~~wherein said master node places~~ placing the boot images and software packages in
~~said software package storage means~~ the first storage and the node
information in ~~said node information storage means~~ the second storage.
31. (currently amended) A computer-readable medium as recited in Claim 21, wherein
[[said]] retrieving [[step]] preferred software version information creates ~~said node's~~
the preferred software version information from said node information storage means
the second storage based on functional features requested by said node.
32. (currently amended) An apparatus of software loading and initialization in a
distributed network of nodes, comprising:
a master node;
~~software package storage means~~ a first storage on said master node for storing
software packages and boot images that the nodes in the network will use ~~be~~
~~using as well as older versions that are kept for regressing a node back to a~~
~~previous boot image or software package version;~~
~~node information storage means~~ a second storage on said master node for storing
preferred software version information[[,]] and node type, ~~and other pertinent~~
information for each node in the network;
~~wherein~~ means for receiving a request for a boot image and software packages from a
node, in the distributed network, that is performing an initial boot requests a
~~boot image and software packages from said master node;~~
means for retrieving, based on the request, said node's preferred software version
information of the node from said node information storage means the second
storage;

means for extracting, using the preferred software version information of the node, a boot image and one or more software packages from ~~said software package storage means using said node's preferred software version information~~ the first storage; and
means for delivering, to the node, the extracted boot image and one or more software packages ~~to said node~~.

33. (currently amended) An apparatus as recited in Claim 32, wherein said node stores the extracted boot image and one or more software packages in its local persistent storage and wherein software version information is extracted from the one or more software packages and stored in the local persistent storage.
34. (currently amended) An apparatus as recited in Claim ~~[[33]]~~ 32, wherein said node, based on a command from said master node, does not store the one or more software packages in ~~[[the]]~~ its local persistent storage device, allowing said master node to download test software packages to said node and temporarily run said node using the test software packages, and wherein when said node reboots, the test software packages will no longer exist on said node.
35. (original) An apparatus as recited in Claim 33, wherein said node reboots and executes the boot image stored in the local persistent storage, and wherein said node verifies the software version information with said master node.
36. (original) An apparatus as recited in Claim 35, wherein if said node has the correct software versions, then said node completes booting by executing the software packages stored in the local persistent storage.
37. (currently amended) An apparatus as recited in Claim 35, ~~wherein~~ further comprising means for retrieving, if said node does not have the correct software versions, ~~said master node retrieves~~ correct software packages from ~~said software package storage means~~ the first storage and ~~[[sends]]~~ sending the correct software packages to said

node, [[and]] wherein said node stores the correct software packages in the local persistent storage and completes booting by executing the correct software packages stored in the local persistent storage.

38. (original) An apparatus as recited in Claim 29, wherein the master node has the ability to categorize nodes into classes where all of the nodes in a particular class of nodes have the same software configuration and may have differing processor types.
39. (original) An apparatus as recited in Claim 32, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.
40. (original) An apparatus as recited in Claim 32, wherein a boot image is customized for a particular type of node and provides basic low-level communications.
41. (currently amended) An apparatus as recited in Claim 32, ~~wherein a user installs~~ further comprising:
means for executing a composite image ~~onto said master node which is executed and creates to create~~ boot images, software packages, and node information[[,]]; and ~~said master node places~~ means for placing the boot images and software packages in said ~~software package storage means~~ the first storage and the node information in said ~~node information storage means~~ the second storage.
42. (currently amended) An apparatus as recited in Claim 32, wherein said means for retrieving [[means]] preferred software version information creates ~~said node's~~ the preferred software version information from ~~said node information storage means~~ the second storage based on functional features requested by said node.
43. (new) A system for software loading and initialization in a distributed network of nodes, the system comprising:
a master node;

a node in the distributed network;
a first storage on the master node, wherein the first storage stores boot images and software packages that nodes in the distributed network will use;
a second storage on the master node, wherein the second storage stores preferred software version information and node type information for each node in the distributed network;
one or more processors on the master node;
one or more sequences of instructions which, when executed by the one or more processors, cause the one or more processors to perform:
 receiving a request for a boot image and software packages from the node that is performing an initial boot;
 based on the request, retrieving preferred software version information of the node from the second storage;
 using the preferred software version information of the node, extracting a boot image and one or more software packages from the first storage; and
 delivering, to the node, the extracted boot image and one or more software packages;
one or more other processors on the node;
one or more other sequences of instructions which, when executed by the one or more other processors, cause the one or more other processors to perform:
 storing the extracted boot image and one or more software packages in local persistent storage of the node;
 extracting software version information from the software packages;
 storing the software version information in the local persistent storage;
 executing the boot image, that is stored in the local persistent storage, to reboot the node.

44. (new) A system as recited in Claim 43, wherein the node, based on a command from said master node, does not store the software packages in the local persistent storage device, allowing said master node to download test software packages to said node and temporarily run said node using the test software packages, and wherein when said node reboots, the test software packages will no longer exist on said node.

45. (new) A system as recited in Claim 43, wherein the one or more other sequences of instructions which, when executed by the one or more other processors, further cause the one or more other processors to perform verifying the software version information with the master node.
46. (new) A system as recited in Claim 45, wherein the one or more other sequences of instructions which, when executed by the one or more other processors, further cause the one or more other processors to perform executing the one or more software packages stored in the local persistent storage to complete booting if the node has the correct software versions.
47. (new) A system as recited in Claim 43, wherein:
the one or more sequences of instructions which, when executed by the one or more processors, further cause the one or more processors to perform retrieving correct software packages from the first storage and sending the correct software packages to the node if the node does not have the correct software versions; and
the one or more other sequences of instructions which, when executed by the one or more other processors, further cause the one or more other processors to perform storing the correct software packages in the local persistent storage and executing the correct software packages stored in the local persistent storage to complete booting.
48. (new) A system as recited in Claim 43, wherein the master node has the ability to categorize nodes into classes where all of the nodes in a particular class of nodes have the same software configuration and may have differing processor types.
49. (new) A system as recited in Claim 43, wherein a software package contains version information, dependency information, and other metadata information pertaining to software in the package.

50. (new) A system as recited in Claim 43, wherein a boot image is customized for a particular type of node and provides basic low-level communications.
51. (new) A system as recited in Claim 43, wherein the one or more sequences of instructions which, when executed by the one or more processors, further cause the one or more processors to perform:
executing a composite image, that a user installs on the master node, to create boot images, software packages, and node information; and
placing the boot images and software packages in the first storage and the node information in the second storage.
52. (new) A system as recited in Claim 43, wherein retrieving the preferred software version information creates the preferred software version information from the second storage based on functional features requested by the node.